We Claim:

1. In an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, a method comprising the steps of:

    providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and

    creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters.

2. The method of claim 1, comprising the further step of:

    providing a view of salient aspects of the code generated by said automatic code generator utilizing the user-selected parameters.

3. The method of claim 2, comprising the further step of:

    changing the user selected parameters for said custom class in said user interface; and

    adjusting the code generated by said automatic code generator to reflect the change in user selected parameters.

4. The method of claim 3, comprising the further step of:

    displaying salient aspects of the adjusted code in said view of code.

5. The method claim 2 wherein said view of code automatically generated includes at least one token, said token being symbolically representative of a non-displayed segment of code.

6. The method of claim 1 wherein said custom storage class declares macros for instances of constant data.

7. The method of claim 1 wherein said custom storage class declares variables for instances of constant data.

8. The method of claim 1 wherein said user-selected parameters control at least one of the manner in which automatically generated code is defined, declared, accessed and addressed.

9. The method of claim 1 wherein said user-selected parameter includes a non-portable directive to a compiler.

10. The method of claim 9 wherein said non-portable directive to a compiler assigns data to at least one memory location in said electronic device.

11. The method of claim 1, comprising the further step of:

creating a separate header file with said automatic code generator in response to the selection of one of said plurality of user-selected parameters.

12. In an electronic device having a modeling and execution environment with at least one graphical model, a system comprising:

a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code from said graphical model ;

a custom storage class in said graphical modeling and execution environment, said custom storage class created utilizing parameters selected by a user from said plurality of selectable parameters; and

a view of salient aspects of code generated by said automatic code generator utilizing the user-selected parameters.

13. The system of claim 12 wherein the user selected parameters for said custom class in said user interface are changed and the source code generated by said automatic code generator is adjusted to reflect the change in user selected parameters.

14. The system of claim 13 wherein the adjusted code is displayed in said view of code.

15. The system of claim 12 wherein said view of code includes at least one token, said token being symbolically representative of a non-displayed segment of code.

16. A medium for use in an electronic device having a graphical modeling and execution environment, said graphical modeling and execution environment including at least one graphical model, said medium holding executable steps for a method, said method comprising the steps of:

    providing a user interface with a plurality of selectable parameters for a custom storage class, said custom storage class specifying the manner in which an automatic code generator creates source code corresponding to data referenced by said graphical model in said graphical modeling and execution environment; and

    creating a custom storage class in said graphical modeling and execution environment utilizing parameters selected by a user from said plurality of selectable parameters.

17. The medium of claim 16, wherein said method comprises the further step of:

    providing a view of salient aspects of code generated by said automatic code generator utilizing the user-selected parameters.

18. The medium of claim 17, wherein said method comprises the further step of:

    changing the user selected parameters for said custom class in said user interface; and

    adjusting the code generated by said automatic code generator to reflect the change in user selected parameters.

19. The medium of claim 18, wherein said method comprises the further step of:

    displaying the adjusted code in said view of code.

20. The medium of claim 17 wherein said view of code automatically generated includes at least one token, said token being symbolically representative of a non-displayed segment of code.

21. The medium of claim 16 wherein said custom storage class declares macros for instances of constant data.

22. The medium of claim 16 wherein said custom storage class declares variables for instances of constant data.

23. The medium of claim 16 wherein said user-selected parameters control at least one of the manner in which automatically generated code is defined, declared, accessed and addressed.

24. The medium of claim16 wherein said user-selected parameter includes a non-portable directive to a compiler.

25. The medium of claim 24 wherein said non-portable directive to a compiler assigns data to at least one memory location in said electronic device.

26. The medium of claim 16, wherein said method comprises the further step of:
        creating a separate header file with said automatic code generator in response to the selection of one of said plurality of user-selected parameters.